6

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

MF

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/959,149	10/28/97	LIMPRECHT	R 3382-47280

TM02/0703

KLARQUIST SPARKMAN CAMPBELL
LEIGH & WHINSTON
ONE WORLD TRADE CENTER SUITE 1600
121 S W SALMON STREET
PORTLAND OR 97204-2988

EXAMINER	
LAO, S	
ART UNIT	PAPER NUMBER
2151	16

DATE MAILED: 07/03/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

PTO-90C (Rev. 2/95)
*U.S. GPO: 2000-473-000/44602

1- File Copy

SM

Office Action Summary	Application No. 08/959,149	Applicant(s) Limprecht, et al
	Examiner S. Lao	Art Unit 2151

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *Apr 11, 2001*

2a) ☒ This action is **FINAL.** 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle* 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-24* is/are pending in the applica

4a) Of the above, claim(s) ______ is/are withdrawn from considera

5) ☒ Claim(s) *22-24* is/are allowed.

6) ☒ Claim(s) *1-7, 11-14, and 18-21* is/are rejected.

7) ☒ Claim(s) *8-10 and 15-17* is/are objected to.

8) ☐ Claims ______ are subject to restriction and/or election requirem

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on ______ is/are objected to by the Examiner.

11) ☐ The proposed drawing correction filed on ______ is: a) ☐ approved b) ☐ disapproved.

12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

a) ☐ All b) ☐ Some* c) ☐ None of:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ______.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

15) ☐ Notice of References Cited (PTO-892)	18) ☐ Interview Summary (PTO-413) Paper No(s). ______
16) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	19) ☐ Notice of Informal Patent Application (PTO-152)
17) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s). 15	20) ☐ Other:

DETAILED ACTION

1. Claims 1-24 are pending. This action is in response to the amendment filed 4/11/2001. Applicant has amended claims 1, 13, 14, 21, 22 and 24.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. Claims 1, 3-7, 11-14, 18-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over The Common Object Request Broker: Architecture and Specification CORBA (Revision 2.0) in view of Ratner et al (U S Pat. 5,889,957) and Steinman ("Incremental State Saving in SPEEDS Using C++").

As to claim 21, CORBA teaches (chapter 4, pages 12-16) server applications (servers, applications), executing (invoke) an application component (object) under control of an operating service (ORB), the application component having a state (context) and function code (method) for performing work responsive to a call (invoke method) from a client (client), destroying the state by the operating service (delete context object by CORBA::CTX_DELETE()). It is noted that a destroyed state in CORBA is not persistent.

CORBA does not teach (1) the step of maintaining, (2) destroying is in response to an indication from application component without action by the client.

As to (1), Steinman teaches (SPEEDS system) maintaining an object state (v1) in main memory between method invocations (between events/messages/method calls, by delta exchange method), see sections 3, 4. Since CORBA and Steinman address object lifetime management (creation/deletion), it would have been obvious to combine the teachings. It is noted that Steinman maintains object state without requiring an indication that work is complete.

As to (2), Ratner teaches in response to an indication from application component (server) without action by the client (message FeOK issued by the server), the operating service (operating system, runtime environment) destroys component (server) state in that the message FeOK issued by the server triggers the breaking of dialog and destruction/deallocation of server state (information maintained in the Dialog Control Block; and a session/dialog is broken immediately after the server sends out a message other than the special error code (which is the code for "continue this session"). See col. 4, lines 50-55; col. 7, lines 28-32; col. 10, lines 17-22; col. 12, lines 2-3.

It would have been obvious that CORBA destroy application component/object state in response to an indication from application component without action by the client as taught by Ratner. The motivation to combine includes that CORBA requires transient states/contexts (page 4-13) and Ratner provides a transient mechanism (context sensitive pathsend), which would have motivated one of ordinary skill in the art to use Ratner in CORBA.

As to claim 1, CORBA teaches (chapter 4, pages 12-16) server applications (servers, applications), executing (invoke) an application component (object) under control of an operating service (ORB), the application component having a state (context) and function code (method) for performing work responsive to a call (invoke method) from a client (client), destroying the state (delete context object via CORBA::CTX_DELETE()).

CORBA does not teach the steps of (1) providing, (2) maintaining, (3) destroying is in response to an indication without action by the client.

As to (1) and (3), Ratner teaches client/server resource management, including providing an interface for the operating service to receive (API) an indication that work is complete (DIALOG ABORT). The API is called by an application component (break a session by the server). See col. 4, lines 50-55; col. 7, lines 28-32; col. 10, lines 17-22; col. 12, lines 2-3. It is noted that completion of work (transaction) is represented by the end of a session/dialogue in Ratner. Ratner further teaches an indication and without client action, as noted in the discussion of claim 21, point (2) with respect to Ratner. Note discussion of claim 21 for motivation to combine.

As to (2), Steinman teaches (SPEEDS system) maintaining an object state (v1) in main memory between method invocations (between events/messages/method calls, by delta exchange method), see sections 3, 4. Since CORBA as modified and Steinman address object lifetime management (creation/deletion), it would have been obvious to combine the teachings. It is noted that Steinman maintains object state without requiring an indication that work is complete.

As to claim 3, CORBA as modified teaches (Steinman) resetting the state (restore state by calling exchange again, section 4).

As to claim 4, CORBA as modified teaches (Ratner) upon a next return following the indication (a session/dialog is broken immediately after the server sends out a message other than the special error code (which is the code for "continue this session"). See col. 4, lines 50-55; col. 7, lines 28-32; col. 10, lines 17-22; col. 12, lines 2-3. It is noted that breaking a dialog in Ratner is accompanied by destruction/deallocation of a server's state (information maintained in the Dialog Control Block).

As to claim 5, note discussion of claims 1 and 4 except for the following limitations.

CORBA teaches (discussion of claim 1) run-time service (ORB) for executing and destroying. CORBA further teaches (chapter 2, page 9, section 2.1.11) instance creation service (object activation), client request (request), return a reference (generate object reference). Typically in CORBA, a client calls object/component functions indirectly by calling a stub which is a form of object adapter to initiate work (invoke object) through the run-time service (ORB, including object adapter) using the reference (stub).

As to claim 6, CORBA as modified teaches (Ratner) calls from client to commit or abort (DIALOG_END, DIALOG_ABORT, callable by client) (col. 6, line 66 - col. 7, line 8; col. 4, lines 36-37, col. 11, lines 66-67). Note rejection of claim 21 for motivation to combine.

As to claim 7, CORBA as modified teaches (Ratner) application component initiates the indication before returning (see discussion of claim 21 with respect to Ratner) and destroying state immediately on return (the message FeOK issued by the server triggers the breaking of dialog and destruction/deallocation of server state (information maintained in the Dialog Control Block; a session/dialog is broken immediately after the server sends out a message other than the special error code (which is the code for "continue this session"). See col. 4, lines 50-55; col. 7, lines 28-32; col. 10, lines 17-22; col. 12, lines 2-3.

As to claim 11, holding a reference and releasing a reference are conventional means for object creation and destruction.

As to claims 12, 14, note the discussion of claims 3 and 4, respectively.

As to claim 13, note discussion of claim 1 and note the equivalence of discarding (claim 13) / destroying (claim 1), and before receiving / without action by the client. CORBA further teaches encapsulating function code (object method) and a processing state for the work in a component (context object), providing a reference (object reference) through an operating service (CORBA) for a client program to call the function code of the component to initiate processing (invoke method) (see discussion of claim 1 with respect to CORBA).

As to claim 18, it is basically a program product claim of claim 5. Note rejection of claim 5.

As to claims 19-20, note discussion of claims 11-12. The factory mechanism of CORBA produces component/object instance and its pointer. When an object is reused, its state is typically reset/reinitialized.

4. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over CORBA in view of Ratner et al and Steinman as applied to claim 1 and further in view of Bishop (U S Pat. 7,765,174).

As to claim 2, Bishop teaches destroying the state (delete object) while retaining a client reference to the object (maintain weak references to object). See col. 1, lines 28-44. Since CORBA and Bishop address object deallocation, it would have been obvious to combine the teachings.

5. Claims 22-24 are allowed.

6. Claims 8-10, 15-17 are objected to as being dependent upon respective rejected base claims, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

7. Applicant's arguments filed 4/11/2001 have been fully considered but they are not persuasive.

Regarding the amended feature of indication from the application component to the operating service at the provided interface of claim 1, applicant appears to argue that the provided interface refers to the disclosed (page 5, lines 31-32) framework-provided interfaces that a component calls. (Remarks, page 8, for example).

The examiner's position is that while this may be disclosed, the argued framework-provided interfaces are not claimed. Therefore, the argument is not persuasive. Further, in Ratner, state destruction is performed by the underlying operating system / runtime environment. And accessing an operating system service is typically through calling a service interface provided by the operating system (API). Thus the provided interface as claimed is met by the API of Ratner.

Applicant further argued that (1) Ratner does not teach without client action because when a dialog is ended by a client or a server, client takes action (Ratner, col. 10, lines 17-23), (2) claim 1 does not recite communication sessions/dialogue. (Page 9, 10).

The examiner disagrees. As to (1), firstly, the client action as claimed is in the context of destroying a component's state, as recited for example in claim 21 *"destroying the state* by the operating service in response to an indication from the application component *without action by the client*," (lines 10-11). In other words, the action of destroying does not require action of the client.

Secondly, the portion of Ratner cited by applicant for support reads "However, if and when an FeOK error code message is issued by the server and received by the client (via Linkmon), not only are linkages (or bindings) 603 and 605 broken, but also linkages 607 are broken, and furthermore, the Dialog Control Block is deallocated and the LACB is marked available." (Col. 10, lines 17-22). Clearly, it is the message FeOK issued by the server that triggers the breaking of dialog and destruction/deallocation of server state (information maintained in the Dialog Control Block). The client action here is to receive the server message FeOK, rather than to trigger breaking of dialog and destruction/deallocation of server state. The cited passage does not teach that the breaking of dialog and the destruction/deallocation of server state is in response to the client receiving the server message. Further evidence of server triggering the break and destruction without client action can be found in Ratner, col. 4, lines 50-55; col. 7, lines 28-32; col. 12, lines 2-3, which teaches the session/dialog is broken immediately after the server sends out a message other than the special error code (which is the code for 'continue this session').

As to (2), communication sessions/dialogue per se is not the teaching relied on. Instead, it is the teaching of an indication of work completion (represented by the end of a session/dialog which is a transaction) that triggers destruction/deallocation of a server's state (information maintained in the Dialog Control Block). Further, it is the combination of CORBA and Ratner that meets the claimed limitation of destroying the state by the operating service in response to an indication from the application component without action by the client. As detailed in the rejection of claim 21, CORBA teaches destroying the state by the operating service but does not teach this occurs in response to an indication from the application component without action by the client. Ratner teaches destruction of a server's state occurs in response to an indication from the application component (server) without action by the client. Therefore, the combined teaching meets the claimed limitation of destroying the state by the operating service in response to an indication from the application component without action by the client.

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for response to this final action is set to expire THREE MONTHS from the date of this action. In the event a first response is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event will the statutory period for response expire later than SIX MONTHS from the date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone number for the organization where this application or proceeding is assigned is (703) 308-9051 for regular communications and After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao
June 25, 2001

[illegible]
MAJID BANANKHAH
PRIMARY EXAMINER